

SQL Server™ 2000 And XML

**Rich Rollman
Lead Program Manager
XML Technologies Team
Microsoft Corporation**

2-322

Microsoft®
PDC 2000
Professional Developers Conference

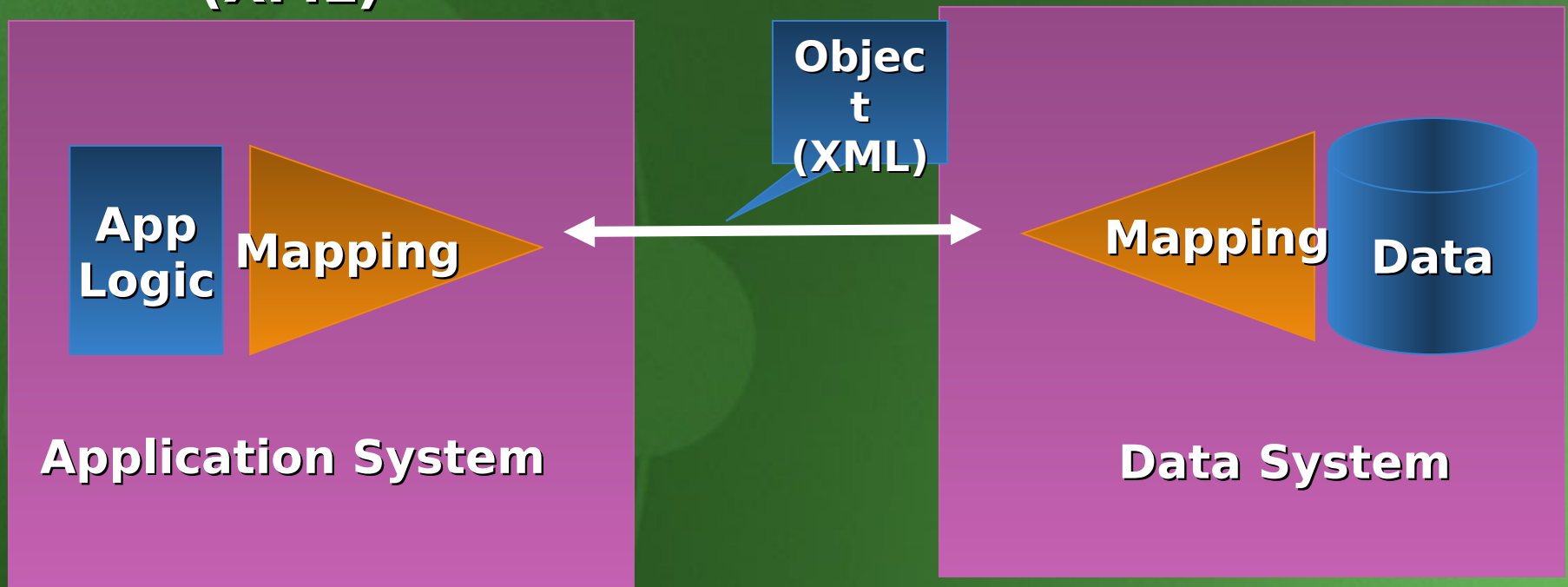
Microsoft®
.net

the defining

point

Loosely Coupled Systems

- Scalable; Many to many
 - Changes in Implementation do not break each other
 - Move data in a standardized format (XML)



Scenarios

- **Web applications with Dynamic Data**
 - **Browser based applications that require data from a database**
- **Business-to-business data processing; messaging**
 - **Data interchange using XML as ubiquitous, extensible, platform independent transport format**

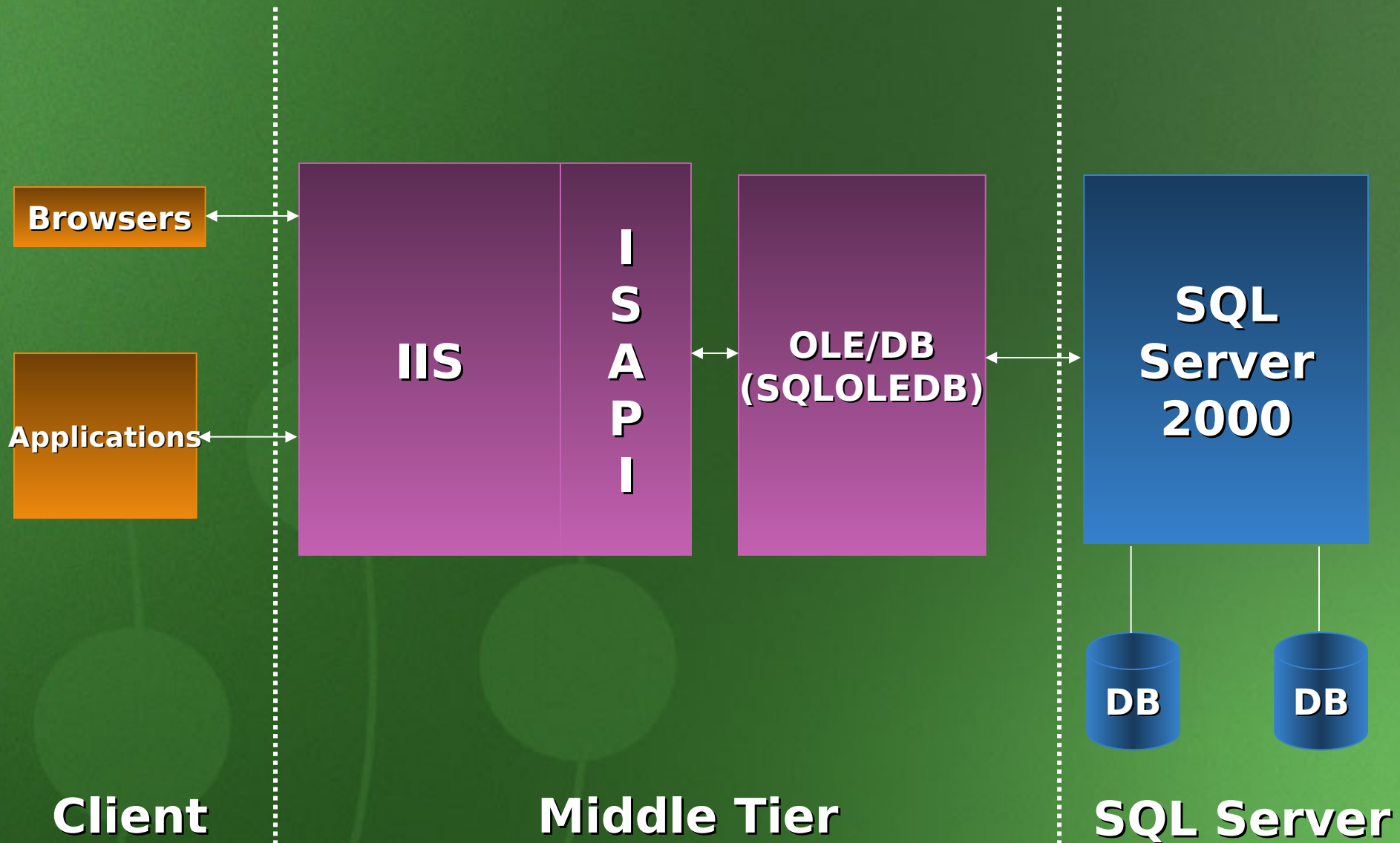
Agenda

- **XML technology primer**
- **HTTP Access**
- **XML Query Results**
- **XML Views**
- **OpenXML**
- **XML Updategrams**
- **Using ADO and OLE-DB (If time)**

XML In 180 Seconds

- XML
 - `<Date>3/2/00</Date>` versus `<P>3/2/00</P>`
- DOM - Document Object Model
- XSLT (XSL)
- DTD
- Schema
 - XML-data reduced
- XPath
 - `/Customer/Order/@OrderID`
- Namespaces

Architecture



HTTP Access

URL types

- **URL Query**

- **http://server/vroot?sql="..."**

- **Direct Query**

- **http://server/vroot/dbobject/xpath**

- **Template**

- **http://server/vroot/vname?params**

- **XML view**

- **http://server/vroot/vname/xpath?
params**

HTTP Access

URL Query



- **http://server/vroot?params**
- **Parameters**
 - **sql=select+*+from+Customers+F
OR+
XML+Auto**
 - **xsl=table.xsl**
 - **contenttype=image/gif**
 - **encoding=UTF-8**
 - **root=root**

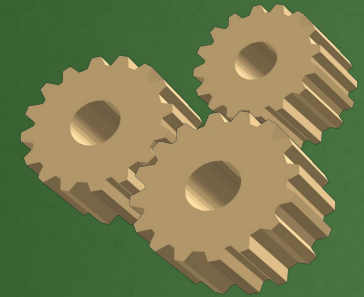
HTTP Access

Direct Query



- **http://server/vroot/dbobject/xpath**
- Singleton select of single column
- Value returned in native format, not XML
- Enables direct retrieval of objects from database
 - Example: Images, OLE objects, etc.
- **XPath Syntax (xpath)**
 - Specifies query in XPath treating tables/views as elements and columns as attributes
 - Example: **http://server/vroot/dbobject/Employees[@EmployeeID=1]/@Photo**

HTTP Access Template



- **http://server/vroot/vname?params**
- **Virtual name (vname)**
 - Direct reference to template file
 - Path to directory
 - **http://server/vroot/vname/customers.xml?params**
- **Two Types of Parameters**
 - Implementation defined
 - **xsl, encoding, contenttype**
 - User Defined
 - Named parameters referenced within the templates

HTTP Access

XML View

- Later...

HTTP Access Security

- Security is set per virtual root
- Permissions are set on SQL Server
- ISAPI DLL has three authentication options
 - Anonymous access
 - Windows® or SQL Server ID/password set directly on virtual root and used for all users
 - Basic authentication
 - Clear text SQL Server ID/password
 - Should be used in conjunction with SSL
 - Integrated security
 - Uses Windows ACLs

HTTP Access

Summary

- **Uses the Internet standard protocol: HTTP**
- **Retrieves data in XML format**
- **Enables data to flow through firewalls**
- **Templates provide:**
 - **Loosely coupled**
 - **Complex queries using single round-trip over Web**

XML Query Results

- SQL Language Extension

SELECT...

FROM...

WHERE...

ORDER BY...

FOR XML (raw |
 auto [, ELEMENTS] |
 explicit)
 [, XMLData]
 [, BINARY base64])

FOR XML

Raw Mode



- One `<row>` element per row in the result set
- No nested elements
- Columns/values in result set are attributes/value on the `<row>`
- Similar to CSV but in XML format

FOR XML

Auto Mode



- **Table/View name in database used for the element name**
- **Column names used for attributes**
 - **ELEMENTS directive produces sub elements instead**
- **Supports nested XML output**
 - **Nesting determined by ordering of columns in SELECT clause**
 - **Sibling relationships not supported**
- **Change names using table and column aliases**

FOR XML

Explicit Mode

- **Provides complete control over format of XML result**
- **Columns can be individually mapped to attributes or sub elements**
 - **Supports arbitrary nesting**
 - **Sibling relationships supported**
- **Collapses/hoists hierarchy**
- **Constructs ID/IDREF relationships**
- **CDATA sections in XML output**
- **Retrieves XML Annotations**

FOR XML Explicit Mode

```
<root>
  <Customer cid="ALFKI">
    <name>Alfreds Futterkiste</name>
    <Order oid="0-10643" />
    <Order oid="0-10692" />
    <Order oid="0-10702" />
    <Order oid="0-10835" />
    <Order oid="0-10952" />
    <Order oid="0-11011"/>
  </Customer>
  <Customer cid="BOLID">
    <name>Bólido Comidas preparadas</name>
    <Order oid="0-10326" />
    <Order oid="0-10801" />
    <Order oid="0-10970" />
  </Customer>
</root>
```

FOR XML Explicit Mode

■ Universal Table

TAG	PARENT	Customer!1!cid!id	Customer!1!name!element	Order!2!oid!id
1	NULL	ALFKI	Alfreds Futterkiste	NULL
2	1	ALFKI	NULL	O-10643
2	1	ALFKI	NULL	O-10692
2	1	ALFKI	NULL	O-10702
2	1	ALFKI	NULL	O-10835
2	1	ALFKI	NULL	O-10952
2	1	ALFKI	NULL	O-11011
1	NULL	BOLID	Bólido Comidas preparadas	NULL
2	1	BOLID	NULL	O-10326
2	1	BOLID	NULL	O-10801
2	1	BOLID	NULL	O-10970

FOR XML Explicit Mode



```
SELECT 1 as TAG,  
       NULL as PARENT,  
       CustomerID AS [Customer!1!cid!id],  
       CompanyName AS [Customer!1!name!element],  
       NULL AS [Order!2!oid!id]  
FROM Customers  
WHERE CustomerID = 'ALFKI' OR CustomerID='BOLID'  
UNION ALL  
SELECT 2,  
       1,  
       Customers.CustomerID,  
       NULL  
       '0-' + CAST(Orders.OrderID AS varchar(32))  
FROM Customers inner join Orders  
ON Customers.CustomerID=Orders.CustomerID  
WHERE Customers.CustomerID = 'ALFKI'  
OR Customers.CustomerID='BOLID'  
ORDER BY [Customer!1!cid!id],[Order!2!oid!id]  
FOR XML explicit
```

FOR XML

Summary

- **Builds on existing SQL query knowledge to generate XML**
- **Provides flexibility to generate arbitrary XML grammars**
- **Can be used**
 - **Directly in URL**
 - **Within a template**
 - **Inside a T-SQL stored procedure**

XML Views

- **Defines an XML View on the database**
- **Uses XML-Data Reduced (XDR) syntax with annotations (mapping schemas)**
 - Similar to DTD but using XML grammar
 - Public schema grammar used by BizTalk
- **Annotations specify the XML to relational database mapping**
 - For column values
 - For relationships between contained tags
- **Uses XPath to query the XML View**

XML Views

Schema

```
<?xml version="1.0" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <ElementType name="Customer" sql:relation="Customers">
    <AttributeType name="ID" />
    <attribute type="ID" sql:field="CustomerID" />
    <element type="Order">
      <sql:relationship key-relation="Customers"
                       key="CustomerID"
                       foreign-relation="Orders"
                       foreign-key="CustomerID"/>
    </element>
  </ElementType>
  <ElementType name="Order" sql:relation="Orders">
    <AttributeType name="OrderID" />
    <attribute type="OrderID" sql:field="OrderID"/>
  </ElementType>
</Schema>
```

XML Views

- XPath Query

- Used with a reference to the mapping schema
- /Customer[@ID="ALFKI"]

- Results in:

```
<Customer ID="ALFKI">  
  <Order OrderID="10643" />  
  <Order OrderID="10692" />  
  <Order OrderID="10702" />  
  <Order OrderID="10835" />  
  <Order OrderID="10952" />  
  <Order OrderID="11011" />  
</Customer>
```


XML Views

URL Query

- `http://server/vroot/vname/xpath?params`
- Virtual name (***vname***)
 - Direct reference to mapping schema
 - Path to directory
 - `http://server/vroot/vname/nwind.xml/xpath?params`
- XPath Query (***xpath***)
 - Specifies set of nodes from virtual document to return
 - Allows specification of predicates
 - Cannot project
- Parameters (***params***)
 - `xsl, encoding, contenttype, root`
 - `http://server/vroot/vname/nwind.xml/xpath?params`



XML Views

Template

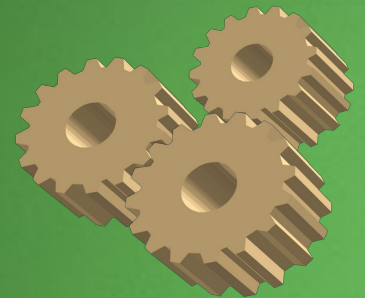
- **http://server/vroot/*vname?*
*params***
- Same as template query with embedded SQL Query except
 - Embedded query references mapping schema file
 - Query is expressed as XPath

XML Views

Template

```
<root xmlns:sql="urn:schemas-microsoft-com:xml-sql"
      sql:xsl="path to XSLT file" >
  <sql:header>
    <sql:param name="state">WA</sql:param>
  </sql:header>

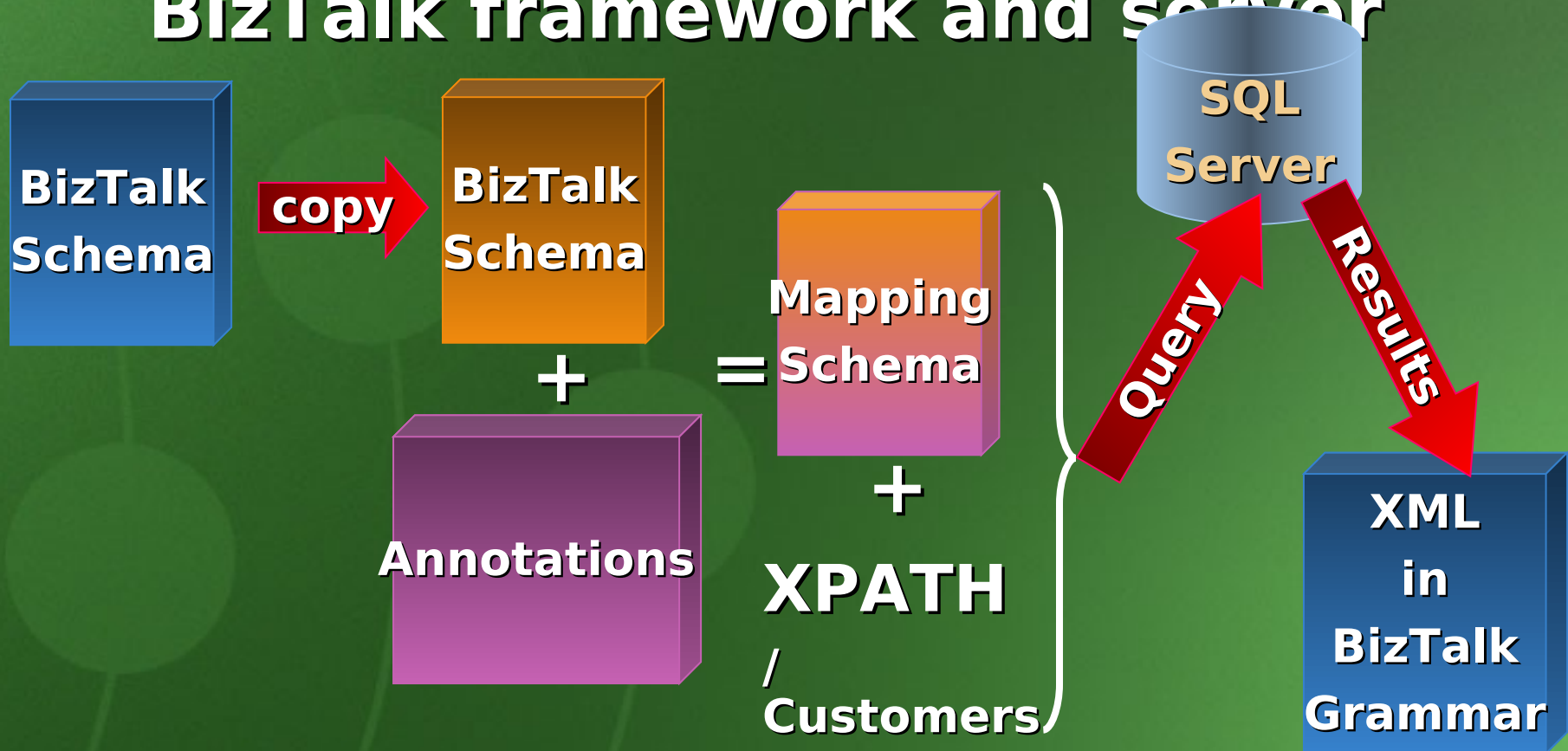
  <sql:xpath-query mapping-schema="nwind.xdr">
    /Customers[@Region=$state]
  </sql:xpath-query>
</root>
```



XML Views

BizTalk

- Uses same schema format as BizTalk framework and server



XML Views

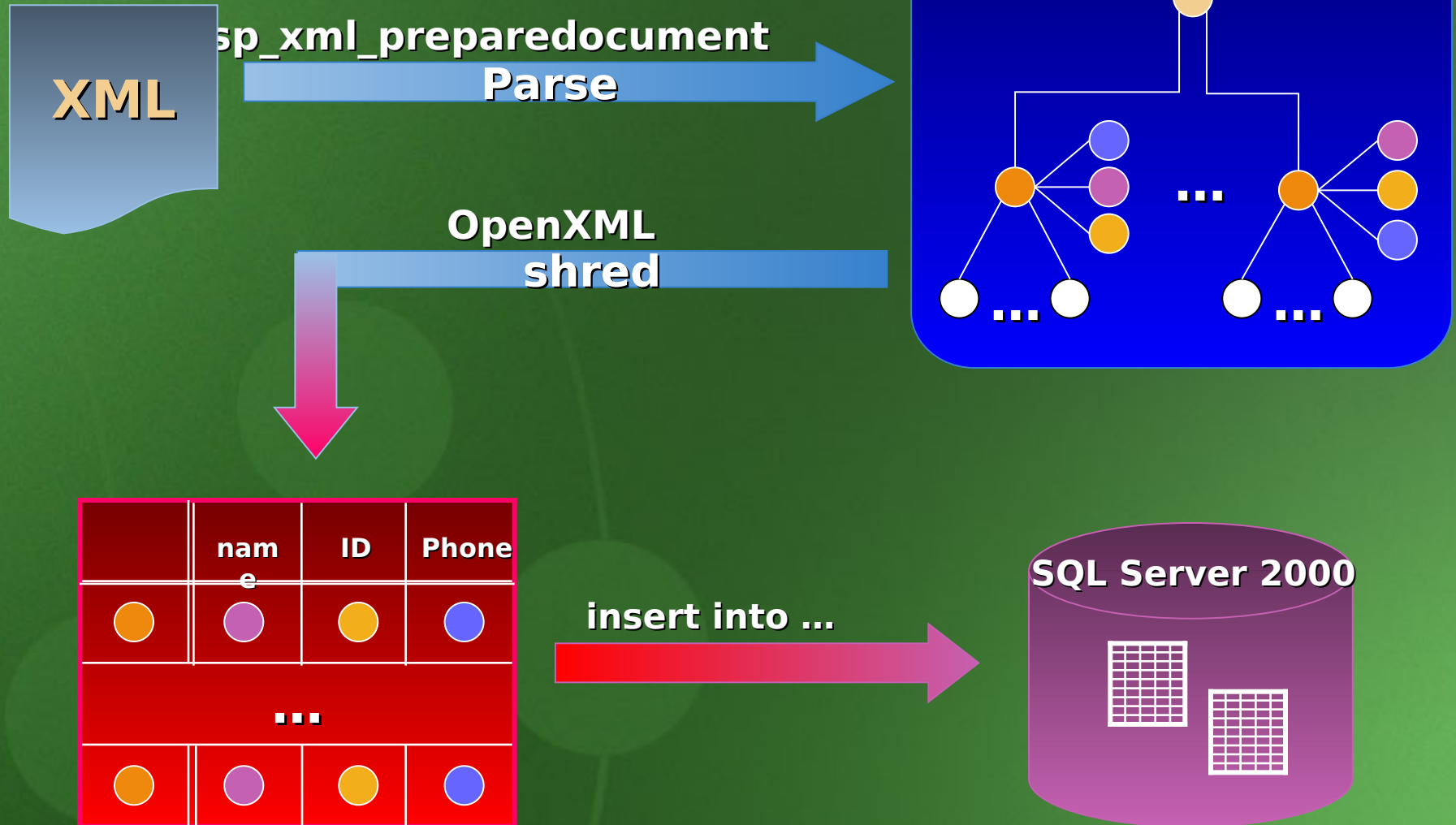
Summary

- **Use XML-centric application programming model**
- **Provides**
 - **Simple, declarative method for defining XML views on the database**
 - **NO CODING REQUIRED**
 - **Mechanism to retrieve XML conforming to Biztalk schema definitions**
 - **XSL for arbitrary output**
- **Integrates with BizTalk**

OpenXML

- **Used with T-SQL Stored Procedures**
- **Provides a relational view on XML**
- **specifies row and column selectors using XPath**
- **Supports**
 - **Attribute and element-centric mappings**
 - **Edge table rowset**
 - **XML annotation/overflow column**

OpenXML

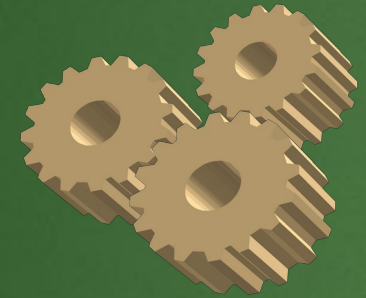


OpenXML

Syntax

- `OpenXML(idoc, rowpattern, flags)`
[WITH (SchemaDecl | Tablename)]
- **Parameters**
 - **idoc** - document handle from `sp_preparedocument`
 - **rowpattern** - XPath selector for rows of view
 - **flags** - Indicates the flavor of mapping to use
- **SchemaDecl** is a schema definition of the form:
`(column_name1 column_type1 [colpattern1], ...,
column_namej column_typej [colpatternj])`
- **Tablename** is the name of an existing table

OpenXML



```
DECLARE @h int
```

```
EXEC sp_xml_preparedocument @h OUTPUT,  
    @xmldoc
```

```
INSERT INTO Customers(CompanyName, CustomerID)
```

```
SELECT CompanyName, CustomerID
```

```
FROM OpenXML(@h, '//Customer')
```

```
WITH (CustomerID varchar(5) '@ID',
```

```
    CompanyName varchar(32) '@name')
```

```
EXEC sp_xml_removedocument @h
```

OpenXML

Summary

- **Leverages existing relational model for use with XML**
- **Provides:**
 - **A mechanism for updating database with data in XML format**
 - **Multi-row updates in single stored procedure call**
 - **Multi-table updates leveraging XML hierarchy**
 - **Queries that join existing tables with XML data**

UpdateGrams

- Expresses modifications to an XML document - not SQL centric
- Uses optimistic concurrency
- Captures inserts, deletes, and updates
- Supports hierarchical updates to multiple tables using mapping schemas
- Can be parameterized
- Planned for Web release - Beta 302000

UpdateGrams

```
<root xmlns:udg="urn:schemas-microsoft-com:xml-updategram">
  <udg:sync mapping-schema="nwind.xml">
    <udg:before>
      <Customer CustomerID="LAZYK"
        CompanyName="Lazy K City Market"
        Address="12 Orchestra Terrace" udg:id="x">
        <Order oid="10482" udg:id="ox"/>
        <Order oid="10545" udg:id="oy"/>
      </Customer>
    </udg:before>
    <udg:after>
      <Customer CustomerID="LAZYK"
        CompanyName="Lazy K Country Store"
        Address="12 Opera Court" udg:id="x">
        <Order oid="10354" udg:id="ox"/>
        <Order oid="10545" udg:id="oy"/>
      </Customer >
    </udg:after>
  </udg:sync>
</root>
```


Using ADO And OLE-DB

- **New dialects on SQL OLE-DB provider**
 - XML
 - XPath
- **New ICommandStream Interface**
- **Returns stream of XML as result of query**
 - Accessible through ADO Stream interface

Using ADO And OLE-DB

■ XML dialect

```
Dim cn As New ADODB.Connection  
Dim cmd As New ADODB.Command  
Dim strmIn As New ADODB.Stream
```

```
cn.Open sConn  
Set cmd.ActiveConnection = cn  
strmIn.Open  
strmIn.WriteText "template"  
strmIn.Position = 0  
Set cmd.CommandStream = strmIn  
cmd.Dialect =  
    "{5d531cb2-e6ed-11d2-b252-00c04f681b71}"  
cmd.Properties("Output Stream").Value = Response  
  
cmd.Execute , , adExecuteStream
```

Using ADO And OLE-DB

■ XPath dialect

```
conn.Open strConn
```

```
Set cmd.ActiveConnection = conn
```

```
cmd.Dialect =
```

```
    "{ec2a4293-e898-11d2-b1b7-00c04f680c56}"
```

```
cmd.CommandText = "/root"
```

```
cmd.Properties("Output Stream").Value = Response
```

```
cmd.Properties("URL Base Path") = "c:\schemas"
```

```
cmd.Properties("Mapping schema") = "nwind.xdr"
```

```
cmd.Execute , , adExecuteStream
```

Futures

- **Web release model**
- **Updategrams**
- **Bulk Load**
- **Support for ODBC Data Sources**
- **SOAP**
- **Enhanced XML Views**
- **DAV**
- **ADO integration**
- **W3C Schemas (XSD)**

Related Sessions And References

- **Other PDC sessions recommended**
 - **2-314 Using the XML Framework**
 - **2-416 XML in Action**
 - **4-331 Using Data and XML in Active Server Pages+**
- **XML Developer Center**
 - **<http://msdn.microsoft.com/xml>**

Where do **you** want to go today?

Microsoft